This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A method of storing a state store on a computing device, the method comprising:

obtaining information from the computing device substantially unique with regard to the computing device, the obtained information being selected from a group consisting of a hardware identification (HWID) generated for the computing device based on one or more identifications of hardware of the computing device, and a specific time associated with the computing device;

generating a specific number of locations on the computing device at which at least a portion of the state store is to be stored;

generating the specific number of pseudo-random file names and the specific number of corresponding paths based at least in part on the obtained information by applying a one-way function to data including the obtained information and employing a resulting output of the function to define the file names[[d]] and the paths, wherein the generated file names and corresponding paths are likewise substantially unique to the computing device;

pairing respective ones of the specific number of generated file names and the specific number of generated paths to form the [[x]]specific number of locations on the computing device; and

storing the state store on the computing device according to the specific number of generated locations on the computing device.

3. (Previously Presented) The method of claim 1 comprising obtaining information specific to the computing device comprising an install time of an operating system thereof.

4. (Previously Presented) The method of claim 1 comprising obtaining information specific to a current period of time, wherein the state store is stored according to a location that varies according to such current period of time

5. (Previously Presented) The method of claim 1 comprising determining the specific number of locations as a number n of parts in which the state store is to be divided times a number m of copies of each part that are to be stored.

6. (Previously Presented) The method of claim 1 comprising generating the specific number of pseudo-random file names, each having a pseudo-random name length.

7. (Previously Presented) The method of claim 1 comprising generating the specific number of paths, each path comprising one of a plurality of levels of an operating system directory path on the computing device.

8. (Previously Presented) The method of claim 1 comprising generating the specific number of paths, each path comprising one of a plurality of levels of a registry path on the computing device.

9. (Currently Amended) The method of claim 1 wherein storing the state store according to the ~~the~~ specific number of generated locations comprises:

protecting the state store by performing at least one of:

signing the state store to produce a signature and appending the signature to the state store; and

encrypting the state store to produce an encrypted state store;

dividing the state store into n parts;

saving each of the n parts m times according to the specific number of formed locations, wherein the specific number = n times m.

10. (Currently Amended) The method of claim 1 wherein storing the state store according to the ~~the~~ specific number of generated locations comprises:

dividing the state store into n parts;

protecting the state store by signing at least one of the n parts of the state store to produce a signature and appending the signature to the part; and

saving each of the n parts according to the ~~the~~ specific number of formed locations.

11. (Currently Amended) The method of claim 1 further comprising retrieving the stored state store, the retrieving comprising:

obtaining the information substantially unique to the computing device;

determining the specific number of locations at which at least a portion of the state store is stored;

generating the specific number of pseudo-random file names and the specific number of corresponding paths based at least in part on the obtained information;

pairing the specific number of generated file names and the specific number of generated paths to form the [[x]]specific number of locations; and

retrieving the state store from the specific number of generated locations.

12. (Previously Presented) The method of claim 11 wherein the state store has been divided into n parts and each of the n parts has been saved according to the specific number of formed locations, and wherein retrieving the stored state further comprises:

retrieving the n parts from the specific number of locations;

reconstituting the state store from the retrieved n parts thereof;

if the reconstituted state store is encrypted, decrypting same; and

if the reconstituted state store is signed to produce a signature, verifying the signature.

13. (Previously Presented) The method of claim 12 wherein the state store has been divided into n parts and each of the n parts has been saved m times according to the specific number of formed locations, and wherein retrieving the stored state comprises reconstituting m copies of the state store from the retrieved n parts thereof, and further comprises randomly selecting one of the m reconstituted copies.

14. (Currently Amended) The method of claim 1 wherein each file name has a length and wherein generating the specific number of pseudo-random file names based at least in part on the obtained information comprises:

hashing data including the obtained information to produce a first hash comprising a string of numbers;

for each file name length, applying a pre-defined serial portion of the first hash to a function to result in the file name length; and

for each Nth file name:

performing a predetermined modification to ~~the~~ an Nth hash;

hashing the modified Nth hash to produce an (N+1)th hash comprising a string of numbers, wherein the first hash is employed to produce a second hash for ~~the~~ a first file name, the second hash is employed to produce a third hash for ~~the~~ a second name, etc.; and

for each file name character of the Nth file name, applying a pre-defined serial portion of the (N+1)th hash to a function to result in the file name character.

15. (Currently Amended) The method of claim 14 wherein each file name length has a preset minimum and maximum, and wherein applying the pre-defined serial portion of the first hash to a function to result in the file name length comprises applying the pre-defined serial portion of the first hash to ~~the~~ a modulo function:

Length = [serial portion mod (maximum – minimum)] +minimum.

16. (Previously Presented) The method of claim 14 wherein applying the pre-defined serial portion of the (N+1)th hash to a function to result in the file name character comprises applying the pre-defined serial portion of the first hash to a conversion table predefined for the computing device.

17. (Previously Presented) The method of claim 14 wherein the modification comprises at least one of a bit shift, a reverse ordering, and a swapping.

18. (Currently Amended) The method of claim 1 wherein each path comprises one of a plurality of levels of an operating system directory path on the computing device, and wherein generating [[x]]specific number of paths based at least in part on the obtained information comprises:

hashing data including or based on the obtained information to produce a path hash comprising a string of numbers;

for each path, applying a pre-defined serial portion of the path hash to a function to result in a level for the path.

19. (Currently Amended) The method of claim 18 wherein each path level has a preset minimum and maximum, and wherein applying the pre-defined serial portion of the path hash to a function to result in the level for the path comprises applying the pre-defined serial portion of the path hash to ~~the~~ a modulo function:

Level = [serial portion value mod (maximum – minimum)] +minimum.

20. (Currently Amended) The method of claim 1 comprising defining successive periods of time, and for each successive period of time:

obtaining information substantially unique to the computing device;

determining a specific number [[x]] of locations at which at least a portion of the state store is to be stored at;

generating the specific number of pseudo-random file names and [[x]]specific number of corresponding paths based at least in part on the obtained information and based at least in part on indicia relevant to the period of time, wherein the generated file names and corresponding paths are likewise substantially unique to the computing device and unique to the period of time;

pairing the specific number of generated file names and the specific number of generated paths to form the [[x]]specific number of locations; and

storing the state store according to the specific number of generated locations, wherein the state store is moved during each successive period of time.

21. (Previously Presented) The method of claim 1 further comprising:

obtaining alternate information relevant to the computing device;

generating at least one pseudo-random file name and at least one corresponding path based at least in part on the alternate information and pairing same to form at least one alternate location; and

storing the obtained information as original information according to the at least one generated alternate location, wherein if the obtained information changes on the computing device, such changed information cannot be employed to retrieve the state store but the alternate information can be employed to retrieve the original information and the original information can be employed to retrieve the state store.

22. (Currently Amended) A computer-readable storage medium for performing a method of storing a state store on a computing device, the method comprising:

obtaining information from the computing device substantially unique with regard to the computing device, the obtained information being selected from a group consisting of a hardware identification (HWID) generated for the computing device based on one or more identifications of hardware of the computing device, and a specific time associated with the computing device;

generating a specific number of locations on the computing device at which at least a portion of the state store is to be stored;

generating the specific number of pseudo-random file names and the specific number of corresponding paths based at least in part on the obtained information by applying a one-way function to data including the obtained information and employing a resulting output of the function to define the file names[[d]] and the paths, wherein the generated file names and corresponding paths are likewise substantially unique to the computing device;

pairing respective ones of the specific number of generated file names and the specific number of generated paths to form the ~~the~~ specific number of locations on the computing device; and

storing the state store on the computing device according to the specific number of generated locations on the computing device.

24. (Previously Presented) The medium of claim 22 wherein the method comprises obtaining information specific to the computing device comprising an install time of an operating system thereof.

25. (Previously Presented) The medium of claim 22 wherein the method comprises obtaining information specific to a current period of time, wherein the state store is stored according to a location that varies according to such current period of time

26. (Previously Presented) The medium of claim 22 wherein the method comprises determining the specific number of locations as a number n of parts in which the state store is to be divided times a number m of copies of each part that are to be stored.

27. (Previously Presented) The medium of claim 22 wherein the method comprises generating the specific number of pseudo-random file names, each having a pseudo-random name length.

28. (Previously Presented) The medium of claim 22 wherein the method comprises generating the specific number of paths, each path comprising one of a plurality of levels of an operating system directory path on the computing device.

29. (Previously Presented) The medium of claim 22 wherein the method comprises generating the specific number of paths, each path comprising one of a plurality of levels of a registry path on the computing device.

30. (Previously Presented) The medium of claim 22 wherein storing the state store according to the specific number of generated locations comprises:

protecting the state store by performing at least one of:

signing the state store to produce a signature and appending the signature to the state store; and

encrypting the state store to produce an encrypted state store;

dividing the state store into n parts;

saving each of the n parts m times according to the specific number of formed locations, wherein the specific number = n times m.

31. (Currently Amended) The medium of claim 22 wherein storing the state store according to the [[x]]specific number of generated locations comprises:

dividing the state store into n parts;

protecting the state store by signing at least one of the n parts of the state store to produce a signature and appending the signature to the part; and

saving each of the n parts according to the specific number of formed locations.

32. (Previously Presented) The medium of claim 22 wherein the method further comprises retrieving the stored state store, the retrieving comprising:

obtaining the information substantially unique to the computing device;

determining the specific number of locations at which at least a portion of the state store is stored;

generating the specific number of pseudo-random file names and the specific number of corresponding paths based at least in part on the obtained information;

pairing the specific number of generated file names and the specific number of generated paths to form the specific number of locations; and

retrieving the state store from the specific number of generated locations.

33. (Previously Presented) The medium of claim 32 wherein the state store has been divided into n parts and each of the n parts has been saved according to the specific number of formed locations, and wherein retrieving the stored state further comprises:

retrieving the n parts from the specific number of locations;

reconstituting the state store from the retrieved n parts thereof;

if the reconstituted state store is encrypted, decrypting same; and

if the reconstituted state store is signed to produce a signature, verifying the signature.

34. (Previously Presented) The medium of claim 33 wherein the state store has been divided into n parts and each of the n parts has been saved m times according to the specific number of formed locations, and wherein retrieving the stored state comprises reconstituting m copies of the state store from the retrieved n parts thereof, and further comprises randomly selecting one of the m reconstituted copies.

35. (Currently Amended) The medium of claim 22 wherein each file name has a length and wherein generating the specific number of pseudo-random file names based at least in part on the obtained information comprises:

hashing data including the obtained information to produce a first hash comprising a string of numbers;

for each file name length, applying a pre-defined serial portion of the first hash to a function to result in the file name length; and

for each Nth file name:

performing a predetermined modification to ~~the~~ an Nth hash;

hashing the modified Nth hash to produce an (N+1)th hash comprising a string of numbers, wherein the first hash is employed to produce a second hash for ~~the~~ a first file name, the second hash is employed to produce a third hash for ~~the~~ a second name, etc.; and

for each file name character of the Nth file name, applying a pre-defined serial portion of the (N+1)th hash to a function to result in the file name character.

36. (Currently Amended) The medium of claim 35 wherein each file name length has a preset minimum and maximum, and wherein applying the pre-defined serial portion of the first hash to a function to result in the file name length comprises applying the pre-defined serial portion of the first hash to ~~the~~ a modulo function:

Length = [serial portion mod (maximum – minimum)] +minimum.

37. (Previously Presented) The medium of claim 35 wherein applying the pre-defined serial portion of the (N+1)th hash to a function to result in the file name character comprises applying the pre-defined serial portion of the first hash to a conversion table predefined for the computing device.

38. (Previously Presented) The medium of claim 35 wherein the modification comprises at least one of a bit shift, a reverse ordering, and a swapping.

39. (Currently Amended) The medium of claim 22 wherein each path comprises one of a plurality of levels of an operating system directory path on the computing device, and wherein generating [[x]]specific number of paths based at least in part on the obtained information comprises:

hashing data including or based on the obtained information to produce a path hash comprising a string of numbers;

for each path, applying a pre-defined serial portion of the path hash to a function to result in a level for the path.

40. (Currently Amended) The medium of claim 39 wherein each path level has a preset minimum and maximum, and wherein applying the pre-defined serial portion of the path hash to a function to result in the level for the path comprises applying the pre-defined serial portion of the path hash to ~~the~~ a modulo function:

Level = [serial portion value mod (maximum – minimum)] +minimum.

41. (Currently Amended) The medium of claim 22 wherein the method comprises defining successive periods of time, and for each successive period of time:

obtaining information substantially unique to the computing device;

determining the specific number of locations at which at least a portion of the state store is to be stored;

generating the specific number of pseudo-random file names and the specific number of corresponding paths based at least in part on the obtained information and based at least in part on indicia relevant to the period of time, wherein the generated file names and corresponding paths are likewise substantially unique to the computing device and unique to the period of time;

pairing the specific number of generated file names and the specific number of generated paths to form the [[x]]specific number of locations; and

storing the state store according to the specific number of generated locations, wherein the state store is moved during each successive period of time.

42. (Previously Presented) The medium of claim 22 wherein the method further comprises:

obtaining alternate information relevant to the computing device;

generating at least one pseudo-random file name and at least one corresponding path based at least in part on the alternate information and pairing same to form at least one alternate location; and

storing the obtained information as original information according to the at least one generated alternate location, wherein if the obtained information changes on the computing device, such changed information cannot be employed to retrieve the state store but the alternate information can be employed to retrieve the original information and the original information can be employed to retrieve the state store.